AFRL-IF-RS-TR-2003-3
Final Technical Report
January 2003

PLAN GENERATION AND EVALUATION USING ACTION NETWORKS

Rockwell Scientific

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. C672

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-3 has been reviewed and is approved for publication.

APPROVED:

FRANK S. BORN
Project Engineer

FOR THE DIRECTOR:

MICHAEL L. TALBERT, Maj., USAF
Technical Advisor, Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
JANUARY 2003

3. REPORT TYPE AND DATES COVERED
Final Sep 95 – Feb 02

4. TITLE AND SUBTITLE
PLAN GENERATION AND EVALUATION USING ACTION NETWORKS

5. FUNDING NUMBERS
C - F30602-95-C-0251
PE - 63226E
PR - C672
TA - 00
WU - 97

6. AUTHOR(S)
Mark Peot

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rockwell Scientific
2530 Meridian Parkway
Durham North Carolina 27713

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFTB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2003-3

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Frank S. Born/IFTB/(315) 330-4726/ Frank.Born@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This effort focused on representation and evaluation of plans accounting for uncertainty in the parameters upon which the plan is based, and also uncertainty in the outcomes that will result from potential actions of the plan. Methods used to accomplish these results included the use of Action Networks, and development of a suite of analysis tools in support of the AFRL Campaign Assessment Tool (CAT - AKA Causal Analysis Tool). Action networks is a language for representing actions and their effects based on Bayesian networks. The analysis tools were intended to provide supporting analysis of air campaign plans including analysis of the value of observing new information and the value of controlling key uncertainties.

14. SUBJECT TERMS
Planning, Uncertainty and Bayesian Networks

15. NUMBER OF PAGES
30

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

1. Overview

The focus of work under this contract has been on the representation and evaluation of plans, accounting for uncertainty of knowledge about the state of the world and about the outcomes of actions.

Action Networks: In the first half of this contract, our proposed plan representation was Action Networks, a language for representing actions and their effects based on Bayesian Networks. The crucial aspect of this representation is that definition of each action is factored according to what aspects of the world state that action depends upon or changes. This factorization is exploited to make plan evaluation, explanation and generation computationally feasible. Our initial conception of Action Networks has been refined to include a more expressive factorization technique, and this new definition has been used as the basis for an algorithm for automated plan generation. We have pursued two forms of plan metrics for plans represented Action Networks: plan evaluation (prediction of expected outcomes) and plan explanation (identification of the important differences between two plans with respect to some prediction). We have developed algorithms for both types of metrics.

We investigated the application of these ideas to Air Campaign Planning. One feature of the Air Campaign Planning problem which has not been reflected in our work is the hierarchical nature of plans and of planning. Action Networks are applicable to "tactical level" actions (such as the expected damage resulting from the employment of a particular aircraft and weapon-type against a particular target, however they do not make use of any hierarchical information, such as is used in the strategy-to-task formalism. We believe that adding hierarchical structure will both extend the usefulness of the Action Network formalism and also provide further computational leverage for evaluation and explanation. Accordingly, we expect that adding hierarchical structure will be a major focus of the second half of this contract.

CAT Analysis and the Effects Based Operations (EBO) Jumpstart Demonstration: In the second half of this contract, we focused our efforts on the development of a suite of analysis tools in support of AFRL's Campaign Assessment Tool (aka Causal Analysis Tool or CAT). We developed three tools: two standalone Visual Basic prototypes and an ActiveX component that was embedded into CAT itself. Each system was intended to provide supporting analysis of air campaign plans, including the analysis of the value of observing new information and the value of controlling key uncertainties. The former is intended to support development of intelligence collection plans. The latter is intended to support identification of key objectives to achieve during the campaign. The plan representation used was an atemporal representation of the causal relationships in an effects-based air campaign plan, relating low level objectives, such as destroying a target to progressively higher objectives.

In the final phases of the program, we developed software to support a larger EBO Jumpstart Demonstration. In this demo, AFRL integrated CAT, CATView (Rockwell's activeX component), the Joint Targeting Toolkit and a University of Oregon Air Campaign scheduler. CAT was used to develop causal models relating target sets to high level objectives along with timing for hitting those objectives. JTT was used to designate specific targets within each target set. The resulting target sets were passed to the air campaign scheduler, which assigned specific aircraft/weapon pairs to targets. Finally, CATView performed analysis to determine the probability that this plan would result in a successful outcome, suggested intel to collect to reduce plan uncertainty, and provided an analysis of the weak points in the plan.

The following sections detail accomplishments in each of the tasks of this contract.

Tasks

1.1 Action Networks

The originally proposed Action Network representation defined actions as fragments of a Bayesian Network which described the effects of actions on fluents (state variables). We have refined this definition to explicitly specify the context-dependence of action effects. Adding this explicit additional representation provides for more economy in the action representation: the "size" of an action (in terms of the number of probabilities required to describe that action) can be substantially reduced in many cases, which in turn provides computational leverage and also makes the specification of actions easier. This tree-factored representation for action under uncertainty has received considerable attention in the academic Artificial Intelligence community. (Figures of simple actions in this representation can be found in the attached documents.)

We have also experimented with a cyclic action representation (a Bayesian network with directed cycles), and have found this representation to be useful in structure-based plan generation.

Action networks represent "primitive" actions, i.e. actions which do not have any further decomposition. A plan is a sequence of actions, which is represented by a single Bayesian Network composed of the network fragments for each action "pasted together" in temporal order. This representation is ideal for simulation or prediction, but did not have any obvious analogue to the hierarchical representation specified by the Strategy-to-Task formalism. We have developed a correspondence between Strategy-to-Task and Action Networks, wherein the lowest level actions ("strategies") in a Strategy-to-Task plan correspond to primitive actions in an Action Network. Higher-level actions in the Strategy-to-Task plan are mapped to sets of lower-level actions, but do not require any explicit representation in the Action Network. This mapping does not intrinsically change the functionality of the Action Network, but the added structure can be exploited for plan manipulation and plan metrics.

Issues that have arisen from this mapping are:

- Strategy-to-Task plans do not require an ordering over actions, while Action Networks require a complete ordering.
- Action specifications in Action Networks have been required to supply complete descriptions of action effects, which is likely to be unrealistic in many cases.
- We intend to address these issues in the second half of the contract.

1.2 Plan Metrics

Literature on air campaign planning refers to plan metrics as measures which determine some aspect of quality of a plan, or which summarize several aspects into some overall score. Our initial intent was to determine some specific metrics and to represent them as utility in Action Networks (thereby allowing a dynamic-programming approach to plan optimization). We are indeed able to specify utility in Action Networks, simply by incorporating utility nodes into the underlying Bayesian network (i.e. making the network an Influence Diagram). Thus, we are able to implement metrics such as probability of goal achievement, or cost or risk minimization.

However, our reading of air campaign literature and review of other work in the ARPI initiative has convinced us that a more valuable service for us to perform is in the area of plan understanding. There are many approaches to plan understanding, but the approaches we have adopted are plan prediction (e.g. simulation) and plan explanation (pinpointing which decisions have the most significant influence on some prediction, and explaining why). Both prediction and explanation have natural interpretations in Action Networks, and both can be defined in terms of evaluation of the underlying Bayesian network. Since Bayesian network evaluation is in the worst-case intractable, we have focused considerable energy on heuristic methods for faster evaluation, especially for approximate or incremental evaluation.

One of the early results of this research was a new algorithm for evaluation called “k-predict.” “K-predict” uses a novel combination of techniques: an approximation technique based on the kappa calculus is used to guess an efficient ordering of cutset instantiations for incremental evaluation of a Bayesian network. A prototype implementation of k-predict performed quite well on a number of test cases.

As a next step, we decided to combine k-predict with “Localized Partial Evaluation,” another heuristic incremental evaluation technique which we believe will work well in complimentary cases to k-predict. In attempting to implement the combination of algorithms, we have designed a much more general architecture for Bayesian network evaluation than has previously been available. Almost every optimization, heuristic, or approximation technique which has been published for Bayesian networks can be expressed within this architecture. We feel that a library based on this architecture will serve as a very robust basis for further research into Bayesian network evaluation techniques. The implementation of this library is approximately half complete.

Plan prediction in Action Networks can be defined quite simply in terms of evaluation of Bayesian networks. Namely, we define plan prediction to be either the estimation of the probability of some events, or the expected value of some quantity, given a plan. “What-if” analysis can be supported by asserting that certain events occur (i.e. entering them as observations in the Bayesian network).

Plan explanation is somewhat more complicated. Several researchers have explored explanation in Bayesian networks, mostly in the context of diagnostic reasoning in medical systems. The general idea of explanation in Bayesian networks is to identify some subset of either the nodes or the paths in the network which have the most impact on some result of interest, and to also “tell a story” about how that influence is carried out (i.e. “filling a cup causes the cup to be full which increases the chance that it may spill"). We have investigated this literature and found that while the general framework is applicable, there are a number of shortcomings in published work with respect to explanation of plans (as opposed to diagnostic systems). We have developed a modified criterion for explanation in Bayesian networks that is appropriate for explaining influences in plans, and most particularly, for explaining the differences between two plans. We are developing an algorithm based on this new criterion that should also be more efficient than previously-published algorithms.

The algorithm that we have been developing to date is based on "flat" Action Networks. Additionally, we are working on an algorithm that uses the hierarchy in a Strategy-to-Task plan to generate more compact and useful explanations.

1.3 Plan Generation

We have constructed an algorithm which we call structure-based plan generation, which uses Action Networks to construct plans. Unlike traditional AI planning systems, the result of structure-based plan generation is a compact representation of *all* possible plans of a given length which can be used to achieve a goal. Structure-based plan generation is able to produce this compact set by exploiting the structure of actions as defined in Action Networks. We have compared the power of this planning paradigm with the power of more traditional AI approaches (see attached documents). A prototype implementation of this algorithm has been written in prolog.

An additional result from our investigation of structure-based plan generation is the application of similar ideas to other graph algorithms, for example the construction of join trees in Bayesian network evaluation. This work resulted in a paper published in AAAI'96.

1.4 Plan Analysis

We developed a COM component (called CATView) that performed several kinds of analysis on effects-based air campaign plans. The input for the system consisted of a Bayes net encoding the causal structure of the air campaign plan and a schedule of air actions.

Analysis techniques supported included:

- Value of information analysis: Determine the most important variables to observe to reduce the uncertainty in a specific objective.
- Control value analysis: Determine the most important variables to control to improve the probability of a selected objective.
- Temporal projection: Determine the probability that a goal will be satisfied as a function of time given the schedule.
- End-state analysis: Determine the probability of various goals and subgoals at the end of an air campaign given the schedule.

1.5 Technology Integration Experiments

We have collaborated informally with the Uncertainty Cluster (University of Washington, Brown, Rockwell) on uncertainty, and with SRI and Klein on plan metrics. For the EBO Jumpstart portion of this program, we worked extensively with the University or Oregon (scheduling and utility-directed search), Advanced Computing Concepts (probability concepts for JTT) and Dr John Lemmer and Mr Simon Vogel of AFRL/IFTB on interfaces to the Campaign Assessment Tool.

2. Plan Analysis

In this section, I will discuss the analysis tools that were developed for CATView, starting with value of information and ending with the value of control.

2.1 Value of Information (VOI)

In order to prioritize ISR objectives, we use the *value of information* [Howard]. In decision theory, the value of information is the difference in expected utility between a decision problem in which the observation is observed prior to making a decision and one in which it is not. The *Expected Value of Perfect Information* is

$$
\begin{aligned}
EVPI = {} & \left(\sum_{O} P\{O\}\left[\max_{A}\sum_{X} P\{X \mid O\}U(A,X)\right] - C(O)\right) \\
& - \max_{A}\sum_{X} P\{X\}U(A,X)
\end{aligned}
$$

where *X* is the unknown state of the world, *A* is the proposed decision, *O* is a potential observation, *C(O)* is the cost of observing O and *U(A,X)* is Cost or value of choosing action *A* if *X* is true.

If C is equal to zero, the EVPI is always greater than or equal to zero. The EVPI is equal to zero only if the observation has no effect on the decision. One instance where this is true is when O is marginally independent of X. In this case,

$$
\begin{aligned}
& \left(\sum_{O} P\{O\}\left[\max_{A}\sum_{X} P\{X \mid O\}U(A,X)\right]\Big]\right) - \max_{A}\sum_{X} P\{X\}U(A,X) \\
= & \left(\sum_{O} P\{O\}\left[\max_{A}\sum_{X} P\{X\}U(A,X)\right]\Big]\right) - \max_{A}\sum_{X} P\{X\}U(A,X) \\
= & \left(\sum_{O} P\{O\}\right)\left[\max_{A}\sum_{X} P\{X\}U(A,X)\right] - \max_{A}\sum_{X} P\{X\}U(A,X) \\
= & (1)\left[\max_{A}\sum_{X} P\{X\}U(A,X)\right] - \max_{A}\sum_{X} P\{X\}U(A,X) = 0
\end{aligned}
$$

2.2 Mutual Information

In order to use EVPI to assess the value of information, we need to assess both the space of decisions and a utility function for each situation/decision pair. Although this is often done for war gaming problems, in practice it is very difficult to assess the utility function or to a priori determine the full set of decision alternatives.

Our goal in using VOI is to provide an ordering over possible information collection activities, as well as to provide a metric to indicate the relative importance of the observation. Empirically, almost any metric on P{X} can be used to provide reasonable orderings on the relative value of observing different variables [Ben-Bassat].

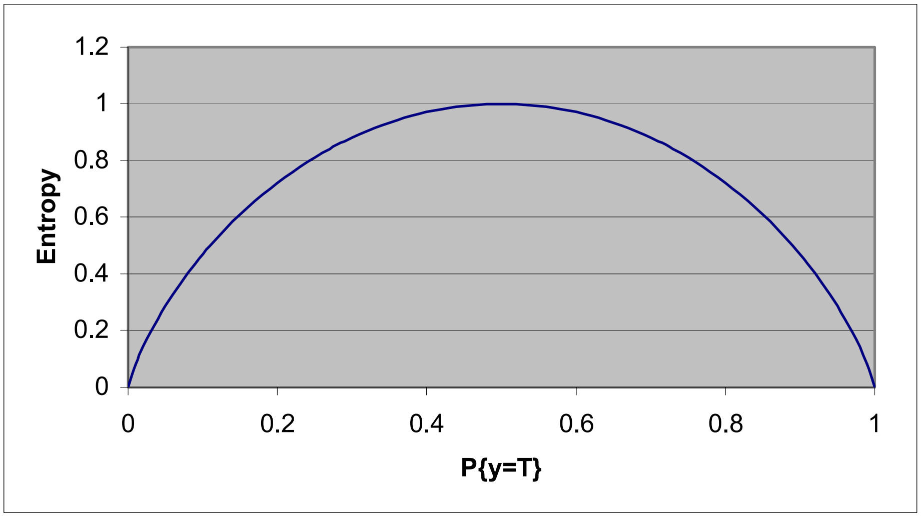
The metric that we use is the *mutual information* between one or more objective variables and the observation variable. The mutual information between *x* and *y* is the expected change in the entropy of *x* given that *y* is observed. Entropy, in turn, is a measure of the amount of "uncertainty" or "randomness" in a random variable. Thus, mutual information measures the expected change in the amount of uncertainty in the variable.

Entropy is given by the following formula:

$$H(x) = -\sum_{x} P\{x\} \lg P\{x\}$$

If *x* is certain, $P\{x\}$ will be 0 or 1 for all values of *x* and the entropy will be 0.[1]
If *x* is uncertain, $H(x)$ will be greater than zero since $1 > P\{x\} \geq 0$ and $\sum_{x} P\{x\} = 1$.

The following figure shows the entropy of a binary variable *y* as the probability that *y* =T is varied between 0 and 1 (the entropy has been scaled to reach a peak of 1.0 when $P\{y = T\} = 0.5$).



Entropy can be calculated for joint distributions, too. The entropy of the joint distribution between *x* and *y* is

$$H(x, y) = -\sum_{x,y} P\{x, y\} \lg P\{x, y\}$$

The mutual information for *x* and *y* is given by:

[1] $\lim_{p \to 0} p \cdot \lg p = 0$ since $\lim_{p \to 0} p \cdot \lg p = \lim_{p \to 0} \left(\dfrac{\frac{d}{dp} \lg p}{\frac{d}{dp} \frac{1}{p}} \right) = \lim_{p \to 0} \dfrac{1 / (p \ln 2)}{-1 / p^2} = \lim_{p \to 0} p^2 / (p \ln 2) = 0$

(L'Hopital's rule).

$$I(x,y)=H(x)+H(y)-H(x,y)\,\textbf{.}$$

Interpreting Mutual Information

Mutual information has a number of interpretations. First of all, it can be derived by considering the expected change in the entropy of *x* given information about *y*:

$$\begin{aligned}
E_y[H(x)-H(x\mid y)] &= H(x)-\sum_y P\{y\}H(x\mid y) \\
&= H(x)+\sum_y P\{y\}\sum_x P\{x\mid y\}\lg P\{x\mid y\} \\
&= H(x)+\sum_{x,y} P\{x,y\}\lg\frac{P\{x,y\}}{P\{y\}} \\
&= H(x)+\sum_{x,y} P\{x,y\}\lg P\{x,y\}-\sum_{x,y} P\{x,y\}\lg P\{y\} \\
&= H(x)+\sum_{x,y} P\{x,y\}\lg P\{x,y\}-\sum_{x,y} P\{y\}\lg P\{y\} \\
&= H(x)-H(x,y)+H(y)
\end{aligned}$$

Equivalently, the mutual information is the expectation of the Kullback-Leibler divergence between the distribution of *x* after observing *y* and the distribution of *x* before observing *y*.

The Kullback-Leibler divergence between distributions P and Q is $D(P\|Q)=\sum_x P\{x\}\lg\frac{P\{x\}}{Q\{x\}}$.

The expectation of the KL divergence between $P\{x\mid y\}$ and $P\{x\}$ is

$$\begin{aligned}
E_y[D(P\{x\mid y\}\|P\{x\})] &= \sum_y P\{y\}D(P\{x\mid y\}\|P\{x\}) \\
&= \sum_y P\{y\}\sum_x P\{x\mid y\}\lg\frac{P\{x\mid y\}}{P\{x\}} \\
&= \sum_{x,y} P\{x,y\}\lg\frac{P\{x,y\}}{P\{x\}P\{y\}} \\
&= -H(x,y)+H(x)+H(y)
\end{aligned}$$

Properties of Mutual Information

Mutual information is an appealing measure of the value of an observation *y* for reducing the uncertainty in *x*. When *x* and *y* are independent, $I(x,y)=0$ since

$$\begin{aligned}
I(x,y) &= H(x)+H(y)-H(x,y)\\
&= H(x)+H(y)+\sum_{x,y} P\{x,y\}\lg P\{x,y\}\\
&= H(x)+H(y)+\sum_{x,y} P\{x\}P\{y\}\lg P\{x\}P\{y\}\\
&= H(x)+H(y)+\sum_{x,y} P\{x\}P\{y\}\lg P\{x\}+\sum_{x,y} P\{x\}P\{y\}\lg P\{y\}\\
&= H(x)+H(y)+\sum_{x} P\{x\}\lg P\{x\}+\sum_{y} P\{y\}\lg P\{y\}\\
&= H(x)+H(y)-H(x)-H(y)=0
\end{aligned}$$

If *y* is equal to *x* (*y* provides perfect information about *x*), then the mutual information between *x* and *y* is equal to the reduction in the uncertainty in *x*, *H*(*x*), when *y* is observed: $I(x,y)=H(x)$. This is easy to see:

$$\begin{aligned}
I(x,y) &= H(x)+H(y)-H(x,y)\\
&= H(x)+H(y)+\sum_{x,y} P\{x,y\}\lg P\{x,y\}\\
&= H(x)+H(y)+\sum_{y} P\{y\}\lg P\{y\}\\
&= H(x)+H(y)-H(y)=H(x)
\end{aligned}$$

2.3 Calculating Mutual Information in CATVIEW

In order to calculate the mutual information between *x* and *every* possible observation y_i in the knowledge base, we need the marginal distributions for *x* and each y_i and the conditional distribution for each y_i given each of the values for *x*.

A Bayesian network is a tool for computing the probability distribution for a *query* variable, *q*, given the values for evidence variables *E*. In order to compute the conditional distribution $P\{y_i \mid x\}$ for a given value *v* of *x*, we set *x* to *v* in the belief network, update the network, and read off $P\{y_i \mid x\}$ from the marginal probability distribution stored in node y_i. Note that we can compute the marginal probability distribution $P\{y_i \mid x\}$ for all of the observation nodes, $\{y_1,\ldots,y_n\}$ given *x* in a single pass through the network. If we store a running total for the joint entropy of *x* and y_i on each node y_i, we can compute the entropy by accumulating the entropy of y_i given *x* weighted by the prior probability of y_i. That is, we

1. set $H(y_i,x)\leftarrow 0$ for each node.
2. accumulate $H(y_i \mid x)P\{x\}$ for each 'instantiation' of x, that is:

```
loop for k = the values of x,
    set x = k in the belief network and update.
    loop for i = the observable variables in the belief network,
        let $H(y_i,x)\leftarrow H(y_i,x)-P\{x\}\sum_{y_i} P\{y_i \mid x\}\lg(P\{x\}P\{y_i \mid x\})$.
```

2.4 Calculating VOI for multiple observations.

The mutual information for multiple observations does not sum. That is:

$$I\left(x,(y,z)\right)\neq I\left(x,y\right)+I\left(x,z\right)$$

This is an important issue because when we are planning the use of ISR assets, we need to consider the impact of all of the information collected by the platform on a wide range of objectives. The sums of the individual mutual information terms is neither an upper bound or a lower bound on I(x, (y,z)). The following example demonstrates why the sum is not an upper bound: suppose that x is equal to y xor (exclusive or) z and both y and z have a prior probability of 0.5. Both y and z are marginally independent of x, so both $I\left(x,y\right)=0$ and $I\left(x,z\right)=0$, yet knowledge of both y and z provides perfect information about x $I\left(x,(y,z)\right)=1$. The sum is not a lower bound either. Suppose that either y or z provide perfect information about x and x has a prior probability of 0.5. Then $I\left(x,y\right)=I\left(x,z\right)=I\left(x,(y,z)\right)=1$.

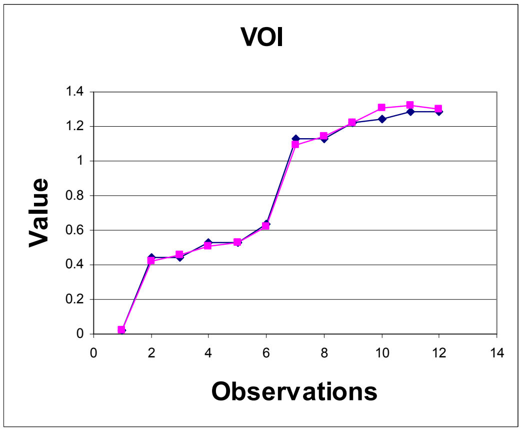
It requires time exponential in the number of observations to compute mutual information exactly. Fortunately, we have found that simulation can be used to derive an effective polynomial epsilon/delta approximation.

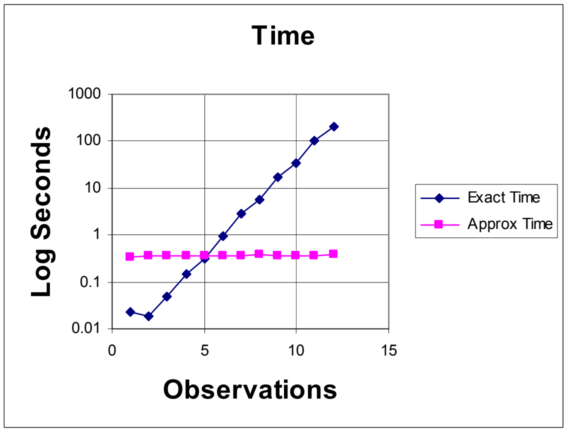
Algorithm: Given O, a vector of observation variables and *x*, a variable of interest, we want to develop an estimate $\hat{I}$ such that $P\left\{I\left(x,O\right)-\varepsilon \leq \hat{I}_N \leq I\left(x,O\right)-\varepsilon\right\} \geq 1-\delta$.

```
Let    $N=\left\lceil \frac{4V_{\max}^2}{e^2}\ln\frac{2}{\delta} \right\rceil$

 and   $\hat{I}=0$

For i = 1 to N
     For j = 1 to |O|
          Draw $o_j$ from $P\{O_j \mid o_1 ,\ldots,o_{j-1}\}$[2]
     next j
     $\hat{I} \leftarrow \hat{I}+H\left(P\left\{X \mid o_1,\mathrm{K}\ ,o_{|O|}\right\}\right)$
next i
$\hat{I} \leftarrow \hat{I}/N$
```

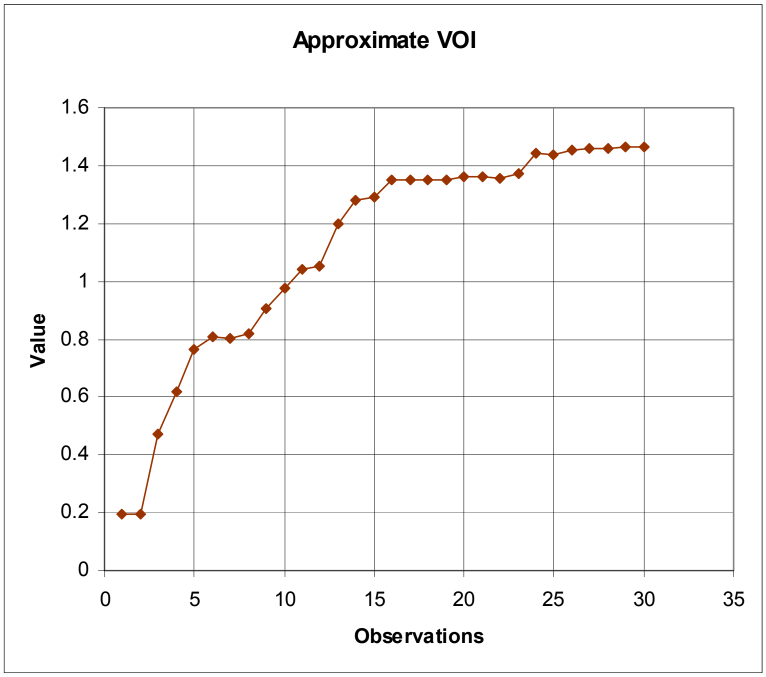
The following figures illustrate the results of a test of the algorithm on a diagnostic Bayes net with approximately 30 observations.

[2] These probabilities are all computed using the Bayes net.





For this example, we used an epsilon of 0.4 and a delta of 0.1. N = 76 iterations. The figure on the left shows the VOI computed for 1 through 12 observations for both the exact algorithm (diamonds) and the approximation algorithm (squares). The figure on the right shows the computation time on a 450 Mhz Pentium processor. Note that the computation time for the approximation algorithm is approximately constant (0.3 seconds) regardless of the number of observations considered. The computation time for the exact algorithm increases exponentially with the number of observations, rising to ~200 seconds to compute the VOI for 12 observations. In our tests, we computed the VOI for all observations in the knowledge base (see below) in under a half second. Computing exact VOI would have required billions of bayes net propagations.

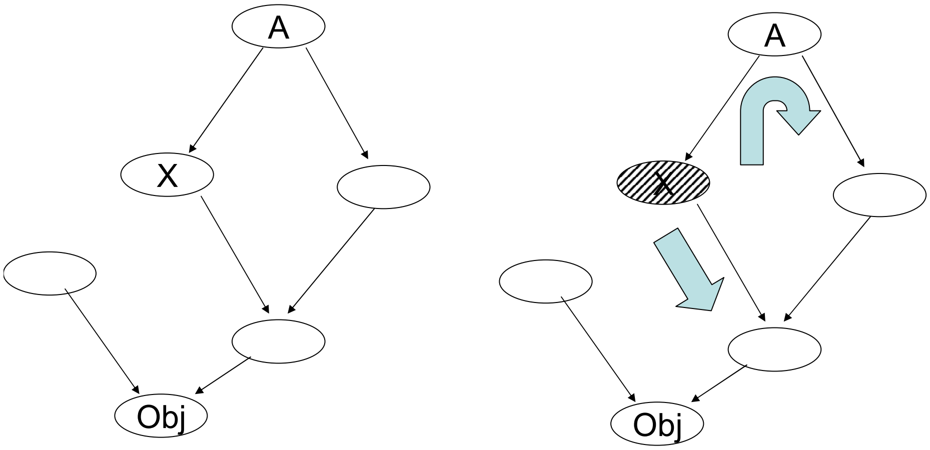


This technique shows that it is practical to use value of information to evaluate the utility of full ISR plans.

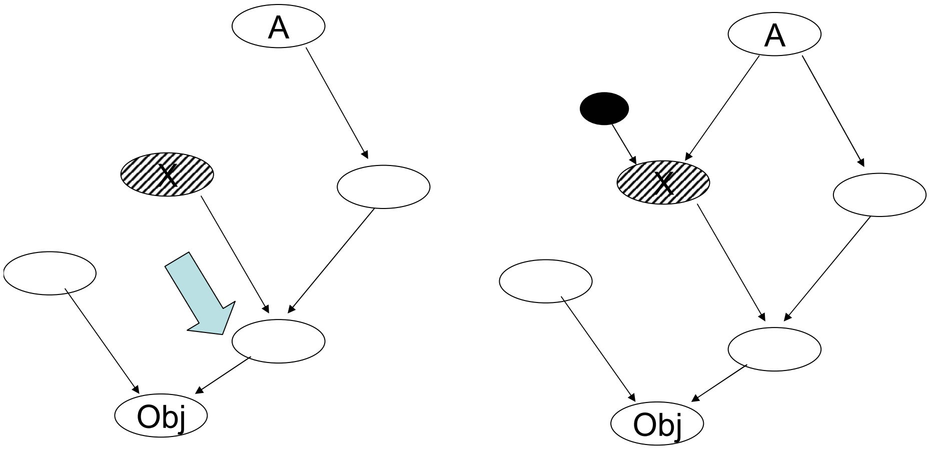
2.5 Value of Control

Value of control measures the extent to which *controlling* and uncertainty will improve a plan. The value of control is the difference in expected utility between a decision scenario in which a variable is set vs. one where it is treated as an uncertainty. Since our objective is to increase the probability of some objective, the value of control for an uncertainty is the amount by which it can increase the probability of a selected objective. In order to compute the value of control, we need to edit the structure of the network.

Iterating over the values for a node X and then selecting the value that has the largest effect on the selected objective Obj, does not work. In the figure below, note that setting X (striped) both influences the probability of downstream nodes, but also changes the probability of upstream nodes, such as A, by acting as a partial observation for their value.

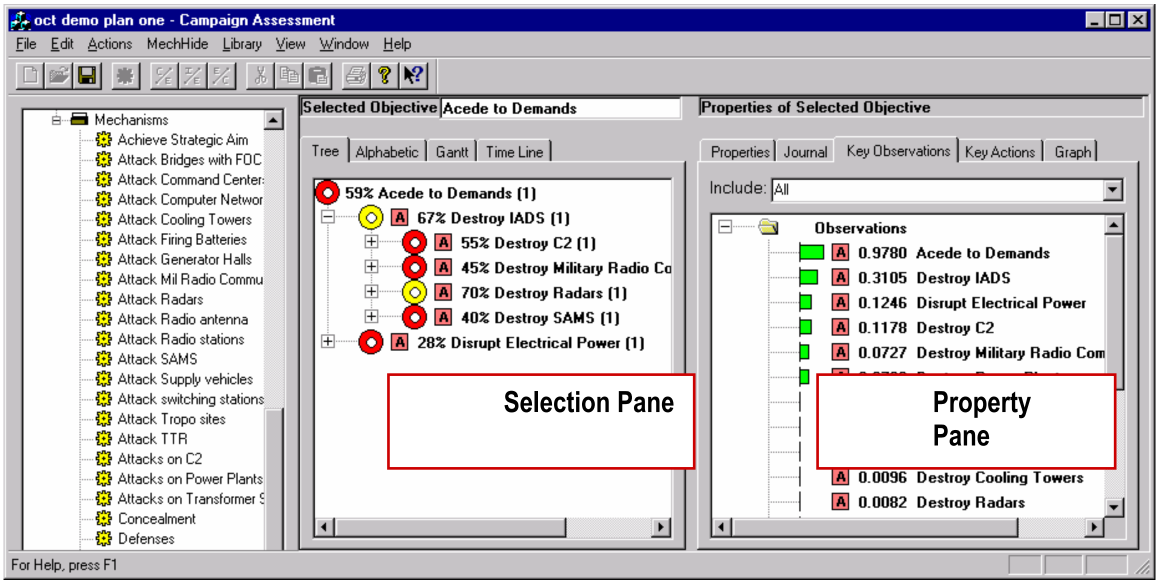


What we need to do is to break the arc between X and its parents before computing the value of control (figure below left). Since interleaving network edits and inference is expensive for most bayes net algorithms, we adopted a canonical form where we use an auxillary variable (black) to effectively disconnect each control variable from its parents. Call this auxillary variable "A". The conditional probabilities for P{X|pa(X), A } when A is true is just equal to the original conditional probability table. When A is false, the probability for X is not a function of its parents (for example, P{X} can be set to a uniform distribution).



2.6 Implementation

The EBO Jumpstart prototype was implemented as an ActiveX component using Microsoft Visual Basic 6.0. The interface is shown below.



oct demo plan one - Campaign Assessment

- Mechanisms
  - Achieve Strategic Aim
  - Attack Bridges with FOC
  - Attack Command Center:
  - Attack Computer Networ
  - Attack Cooling Towers
  - Attack Firing Batteries
  - Attack Generator Halls
  - Attack Mil Radio Commu
  - Attack Radars
  - Attack Radio antenna
  - Attack Radio stations
  - Attack SAMS
  - Attack Supply vehicles
  - Attack switching stations
  - Attack Tropo sites
  - Attack TTR
  - Attacks on C2
  - Attacks on Power Plants
  - Attacks on Transformer S
  - Concealment
  - Defenses

Selected Objective: Acede to Demands

- 59% Acede to Demands (1)
  - A 67% Destroy IADS (1)
    - A 55% Destroy C2 (1)
    - A 45% Destroy Military Radio Co
    - A 70% Destroy Radars (1)
    - A 40% Destroy SAMS (1)
  - A 28% Disrupt Electrical Power (1)

Properties of Selected Objective

Include: All

- Observations
  - A 0.9780 Acede to Demands
  - A 0.3105 Destroy IADS
  - A 0.1246 Disrupt Electrical Power
  - A 0.1178 Destroy C2
  - A 0.0727 Destroy Military Radio Com
  - A 0.0096 Destroy Cooling Towers
  - A 0.0082 Destroy Radars

For Help, press F1

The component occupies the right 2/3 of the screen (the remaining items on the screen are menus and dialogs belonging to the AFRL Campaign Assessment Tool). The left half of the tool presents alternative views of the entire plan. The pane on the right displays properties or analyses for nodes selected in the selection pane.

The Selection Pane presented four alternative views:

- Probability Tree View: A tree of all actions and objectives with probabilities.
- Alphabetic View: an alphabetic list of all actions,
- Gantt View: a Gantt chart showing the current schedule and schedule constraints.
- Time Line: a table that allows the user to sort all actions by start or end time.

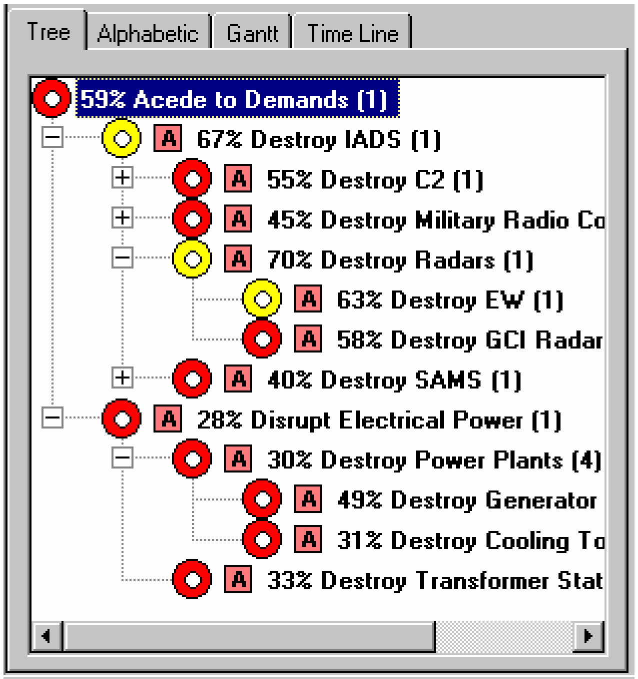
The Property Pane presented three analysis tools and two properties pages.

- Properties: presents all information associated with the definition of the action.
- Journal: Presents a log of all observations made thus far.
- Key Observations: Show a list of the key observations to make in order to maximally reduce uncertainty in the selected variable. The selected variable is always at the head of the list to show the value of a perfect observation.
- Key Actions: Show a list of the key places to intervene in order to maximally increase the probability of success for the selected objective.
- Graph: Show the probability of success for the selected objective as a function of time.

2.6.1 Selection Pane

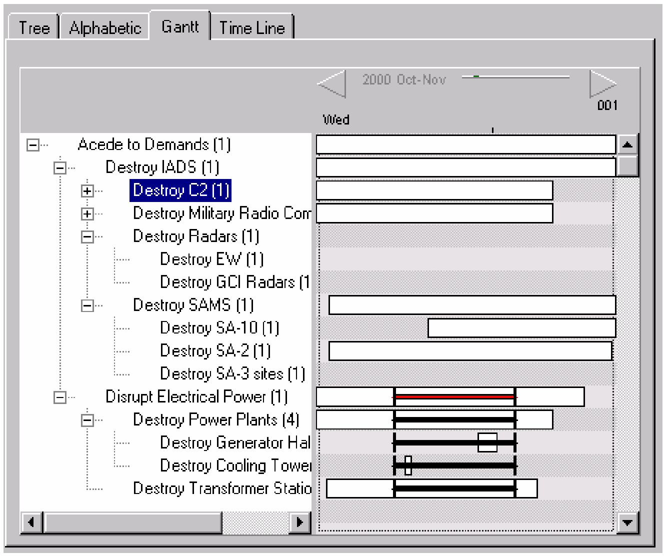
All of the tools in the selection pane allow the user to select and set properties for selected variables. When a variable is selected, the tool computes and caches various analyses and displays the results in the Properties Pane (see below). As observations are made or actions are executed, the user can enter the results of those actions and observations and immediately see the impact on the probability of success for the plan.

The Probability Tree View presents the tree of actions from the highest level objective (Acede to Demands) all of the way down to the target level. The probability numbers in the tree view show the projected probability of success for each objective given the current schedule. Red, Yellow and Green stop lights were set depending on the probability of the objective, allowing the user to quickly drilldown to determine the source for an execution or planning problem.   All of the figures in this section use the same model, a causal model of the Kosovo campaign assembled by Dr. Maris “Buster” McCrabb for the Effects Based Operations Jumpstart demonstration. The overall objective of the plan is to get Milosevic to accede to UN demands. If he does not, the plan will be to put pressure on him by stripping him of air sovereignty. If this does not work, the plan is to turn out the lights.


Tree | Alphabetic | Gantt | Time Line

59% Acede to Demands (1)
- 67% Destroy IADS (1)
  - 55% Destroy C2 (1)
  - 45% Destroy Military Radio Co
  - 70% Destroy Radars (1)
    - 63% Destroy EW (1)
    - 58% Destroy GCI Radar
  - 40% Destroy SAMS (1)
- 28% Disrupt Electrical Power (1)
  - 30% Destroy Power Plants (4)
    - 49% Destroy Generator
    - 31% Destroy Cooling To
  - 33% Destroy Transformer Stat

The Gantt View presented the schedule and constraints and allows the user to adjust temporal constraints. White bars represent the full temporal extent of actions or sets of actions. The red bars represent user-settable constraints and the black bars represent constraints that are implied by other constraints. In the example below, the commander is attempting to force the scheduler to destroy the power plants at a particular time during the mission.


Tree | Alphabetic | Gantt | Time Line

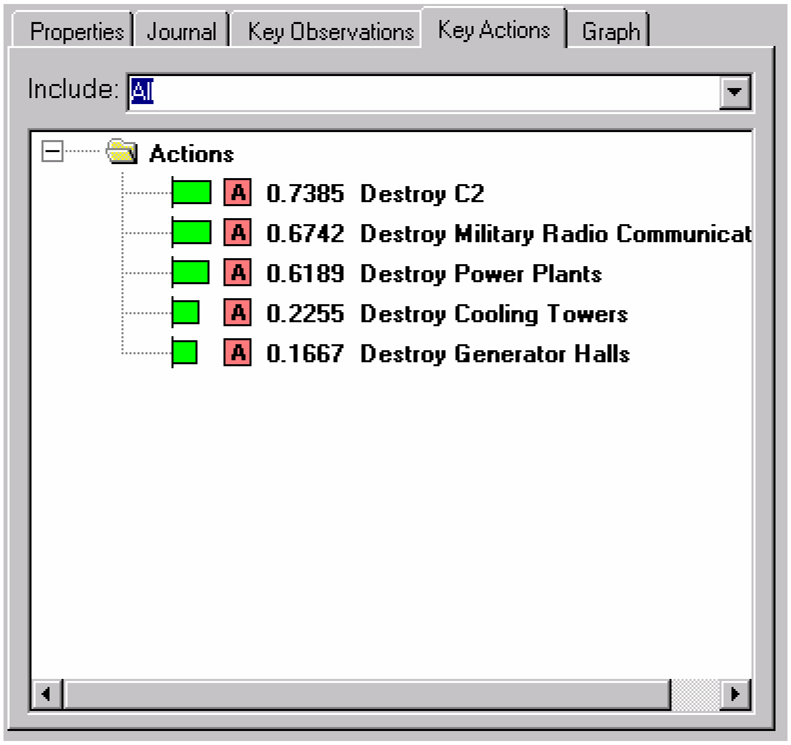
2000 Oct-Nov
001
Wed

Acede to Demands (1)
- Destroy IADS (1)
  - Destroy C2 (1)
  - Destroy Military Radio Com
  - Destroy Radars (1)
    - Destroy EW (1)
    - Destroy GCI Radars (1
  - Destroy SAMS (1)
    - Destroy SA-10 (1)
    - Destroy SA-2 (1)
    - Destroy SA-3 sites (1)
- Disrupt Electrical Power (1)
  - Destroy Power Plants (4)
    - Destroy Generator Hal
    - Destroy Cooling Towe
  - Destroy Transformer Statio

2.6.2 The Properties Pane

The Properties Pane shows properties or analyses for the selected action or objective.

Key Actions illustrates the result of the value of control computations. For this analysis the user selected "Destroy C2" as the objective. The display lists all of the variables that can increase the probability that C2 is destroyed in order of increasing effect on this variable. The number to the left of each variable is the increase in probability due to accomplishing the action. For example, if you could somehow destroy the enemy's ability to communicate, you would increase the probability that C2 were destroyed from 0.2615 to 1.0, a gain of 0.7385. On the other hand, if you destroyed just the generator halls, you would increase the probability that C2 was destroyed by 0.1667.



Include: All

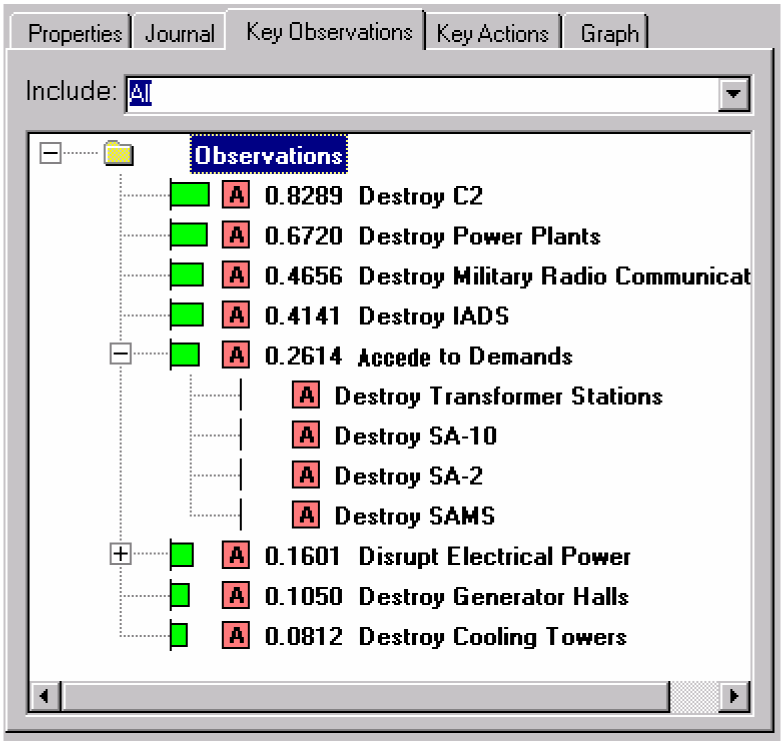
Actions

🟩 A 0.7385 Destroy C2
🟩 A 0.6742 Destroy Military Radio Communicat
🟩 A 0.6189 Destroy Power Plants
🟩 A 0.2255 Destroy Cooling Towers
🟩 A 0.1667 Destroy Generator Halls

Key Observations (next page) shows the value of information for all variables that reveal information about the selected objective. The units for VOI are in bits of entropy. If you could directly observe whether C2 were destroyed, this fact would provide 0.8289 bits of information. On the other hand, if you just observed whether electrical power was disrupted, this would decrease your uncertainty in whether C2 were destroyed by 0.1601 bits.

Note that fairly indirect observations provide data on whether C2 is destroyed. For example, knowing whether Milosevic acceded to UN demands provides evidence that the C2 system was, in fact, destroyed.

The Key Observations panel is a tree. Some observations are only relevant if other variables are observed. For example, in this model, determining whether the SAM system has been destroyed reveals information about "Destroy C2" IF we observe whether Milosevic has acceded to UN demands.

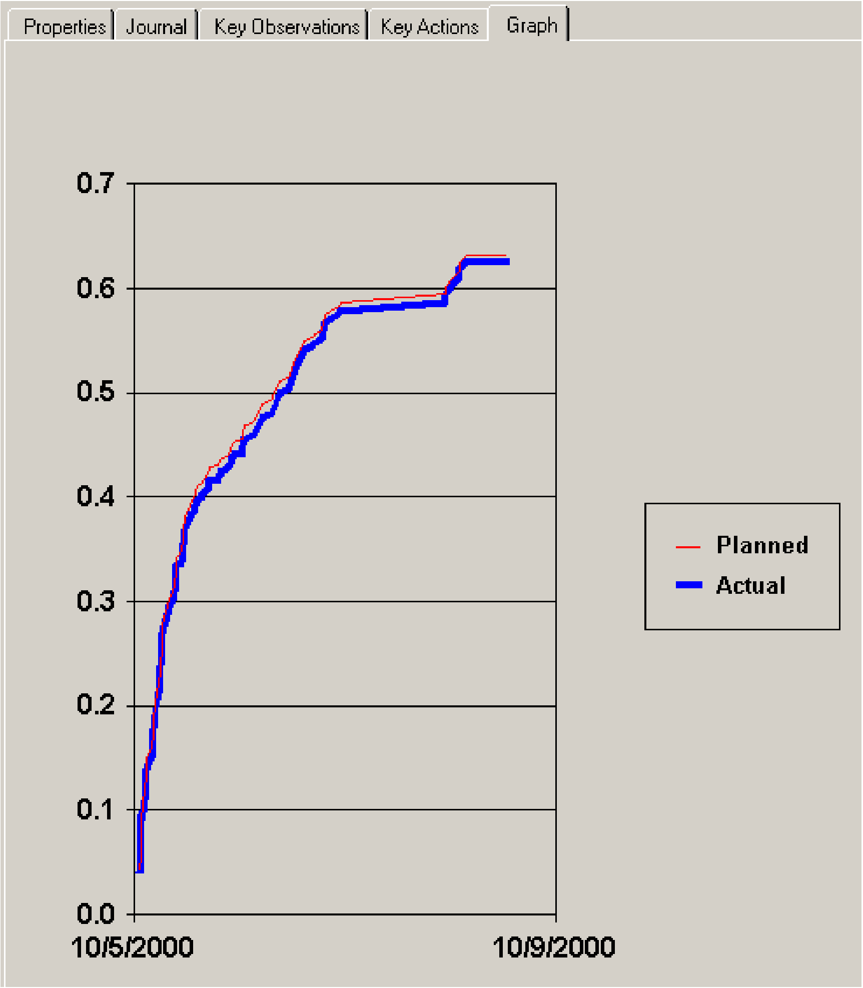


Include: All

Observations

- 🟩 A 0.8289 Destroy C2
- 🟩 A 0.6720 Destroy Power Plants
- 🟩 A 0.4656 Destroy Military Radio Communicat
- 🟩 A 0.4141 Destroy IADS
- 🟩 A 0.2614 Accede to Demands
  - A Destroy Transformer Stations
  - A Destroy SA-10
  - A Destroy SA-2
  - A Destroy SAMS
- 🟩 A 0.1601 Disrupt Electrical Power
- 🟩 A 0.1050 Destroy Generator Halls
- 🟩 A 0.0812 Destroy Cooling Towers

The final pane (next page) shows the projected probability of success for an objective (in this case "Accede to Demands" as a function of the current air campaign schedule. This graph shows the cumulative effect of all elements of the air campaign plan over time.



3. Action Networks

The action networks portion of the contract is fully documented in previously published papers.

Draper, Denise, “Plan Explanation and Explanation in Bayesian Networks”, report, July 1996.

Boutilier, Craig and Goldszmidt, Moises, “The Frame Problem and Bayesian Network Action Representations”, in the Proceedings of the Canadian Conference for Artificial Intelligence, 1996.

Boutilier, Craig, Friedman, Nir, Goldszmidt, Moises and Koller, Daphne, “Context-Specific Independence in Bayesian Networks”, in Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, pp 115-132, 1996.

Darwiche, Adnan, “Utilizing Knowledge-Base Semantics in Graph-Based Algorithms”, in the Proceedings of the Thirteenth National Conference on Artificial Intelligence, pp 607-613, 1996.

Background Study: Plan Explanation and Explanation in Bayesian Networks

Denise L. Draper

July 1996

4. Plan Explanation

4.1 What is Plan Explanation?

You've got a plan, say an air attack plan including target selection down to support operations to get everything to the right place at the right time. How does the plan fit together? If some particular operation is not completed on time, what other activities will be delayed? What effect would that delay have on their success, or on the outcome of the plan as a whole? Suppose a plan evaluator tells you that the plan is fragile with respect to changing weather conditions. How is it fragile? If the weather were stormy instead of clear, which activities would be affected? Or suppose part of the plan uses a particular resource—a set of transport aircraft, perhaps—and some other military operation also wants that resource. Can you substitute a different resource, or reschedule its use, without disrupting the plan? Why was that resource being used anyway?

If we define the general subject of *Plan Understanding* as comprising any kind of procedure that helps an individual to understand a plan—how it works, how good is it, what happens if you change part of it—then we can distinguish two subcategories of plan understanding:

1. Procedures that help an individual to understand what is in a plan: what activities are involved, what order do they take place in, etc. These kinds of procedures we will call *Plan Visualization*. A common example of a plan visualization mechanism is a Gantt chart, which shows the ordering of different activities.
2. Procedures that help an individual to understand why something is part of a plan, or how a plan achieves certain goals, etc. These are the kinds of procedures that we will call *Plan Explanation*. Many formal requirements analysis or work flow models incorporate some kind of explanatory mechanism.

To some extent, this dichotomy is arbitrary—any realistic use of large-scale plans will require both capabilities, and they will probably be inter-linked. But answering questions about "why" or "how" typically involves a deeper understanding of the domain and how the entities involved (activities, constraints) interact with one another.

In order to be able to explain something about a particular plan, we generally have to start by knowing not only what the plan is, but also what it is supposed to achieve, what constraints it must satisfy, etc. Plan explanation then usually consists of determining how certain activities or constraints in the plan cause (or fail to cause) goals to be achieved (or required constraints to be satisfied, etc.). In this regard, we can see that plan explanation is closely related to plan evaluation: where plan evaluation takes a plan and returns some measure of how good the plan is (according to some criterion), plan explanation returns (some part of) the line of reasoning justifying how the measurement was arrived at.[3]

[3] In fact, in the field of automated diagnosis, the primary purpose for explanation is as a justification of an automatically generated diagnosis (which is roughly analogous to a plan evaluation). In experiments with medical diagnosis systems, medical students were more likely to trust the diagnosis generated by an automated system if an

To reiterate, the kinds of questions that plan explanation would be used to answer are:

- How does a plan achieve its objectives, or meet certain criteria?
- How would a change in conditions affect a plan?
- Why are certain activities included in a plan?

These questions concern only a single plan. But often we will be interested in more than one plan, in which case the focus usually shifts away from understanding the structure of how one plan works towards understanding how two plans differ, and what the consequences of those differences are. In other words, we would also like to be able to answer questions such as:

- Why is plan X better than plan Y (according to some specific evaluation criterion)?
- Under what conditions or assumptions would plan X be better than plan Y?

4.2 Two General Approaches to Plan Explanation

There are two fundamentally different techniques for plan explanation. The first is usually called "rationale capture," and consists of asking the decision makers to record reasons for their decisions—in other words, for every decision made, record why it was made the way it was, the assumptions that were used, and the anticipated overall effect on the success of the plan. Plan explanation is then a process of sifting through the recorded rationale for those relevant to the question at hand.

Another technique, employed in classic AI planning as well as in Decision Analysis, is to construct a formal model describing the effects of individual actions[4]—resources the action requires, how the action changes the world (which can be different in different situations), and so forth. A formal model could be logical, or probabilistic, or fuzzy, etc. Given a formal action model, the appropriate inference algorithm can be used to automatically infer properties of the plan as a whole from the properties and interactions of the individual actions that make up the plan.

The information stored in rationale capture is generally less formal than the action models used in AI techniques. This can be either a strength or a weakness. The advantage of informality is that decision makers can put in arbitrary information that they believe will be useful—for example, informal rationale can address political or psychological issues that would be difficult or impossible to model formally. Formal action models, in contrast, require that the scope and structure of the rationale be fixed in advance, and limited to those aspects that can be formally modeled. Moreover, creating formal action models is a time consuming and costly task.

On the other hand, by establishing a formal action model, it is possible to "chain together" information from individual actions to determine indirect effects on the plan as a whole. With informal rationale this sort of combination is generally not possible to do accurately.

It may be feasible to combine aspects of both approaches, either by allowing informal rationale in addition to a formal action model, or by constructing the formal action model *in situ*, as the plan itself is constructed. In either case, we would expect the hybrid system to be more flexible, but also probably to suffer from greater inconsistencies (which could lead to erroneous or misleading explanations).

explanation or rationale for the diagnosis was also given. Probably even more important, the use of explanation also increased the confidence of the medical students to *disagree* with an automatic diagnosis, when they could see that the rationale did not adequately account for certain aspects of the case. [Suermondt & Cooper, 1992]

[4] We use the terms "operation" or "activity" interchangeably to refer to any part of a plan considered as a unit, and the term "action" to refer specifically to a formally-modeled operation or activity.

4.3 The Main Issue in Plan Explanation: Too Much Information

A difficulty that arises with plan explanation techniques is restricting the amount of information given in the answer to a question. The problem is that the functioning of well-constructed plans is often very organic—every part is intimately related to many other parts. The entire plan may seem to be its own best explanation.

There are a number of strategies to moderate the amount of information presented in an explanation:

1. Filter out parts of the explanation that have less significant impact on the plan (for example, small time delays, or small probability of occurrence).
2. If plans are hierarchical (having higher level activities which are expanded into sets of lower level activities), explanation can concentrate on the more abstract levels when details from the lower level activities are not necessary.
3. Often, there are parts of a plan which are crucial to its success, but are also in some sense routine. For example, it is crucial that a patient undergoing surgery be present at the hospital before the surgery, but any reasonable plan would have gotten the patient to the hospital somehow. In other words, when we try to understand even a single plan in isolation, we are usually implicitly comparing that plan to some set of "reasonable alternatives," and we usually are only interested in the significant ways in which a plan differs from those alternatives. [Note: this may not be true if the point of understanding a plan is to learn the planning domain; learning what is routine can also be quite important.]

Implementing any of these strategies involves solving some technical problems. One of the primary issues is determining the significance of one piece of information with respect to the overall explanation. In a rationale capture system, it would be possible to directly include information about significance (for example, listing those assumptions about timing, conditions, etc. that are most crucial for success of an operation), but it would be difficult to combine these assessments (for example, weather might have only a minor impact on a number of operations, considered individually, but the interaction between those impacts might be significant—in an informal system, this would be difficult or impossible to detect automatically). In formal action models, significance can be formally defined and measured (in probabilistic models it is closely related to sensitivity analysis), but it is computationally expensive to compute (algorithms for probabilistic models are exponential). If, as is likely, it proves too expensive to compute significance measures for plans of realistic size, then greedy heuristic approaches or approximation algorithms will need to be developed.

The third strategy listed above also requires the ability to determine which parts of a plan are "routine," which in turn requires some knowledge of what the set of reasonable alternatives is. When comparing plans, the set of alternatives is obviously given by the plans one is comparing, and the problem reduces to that of measuring the significance of the differences between plans. Also, in Decision Analysis (i.e. Influence Diagrams), alternatives are given explicitly, so the problem is again trivial. But explicitly listing alternatives is not generally practical for large-scale planning, so solving this problem for explanations of a single plan remains an open issue. One possible approach to this problem is to exploit the use of hierarchy in a hierarchical planning system, attaching approximate or vague behaviors, to high-level activities in the plan, which are intended to indicate "normal" behaviors for reasonable implementations of that activity, and using the approximations as a basis for comparison.

Supposing these problems are solved, there still remains the issue of how much information to present in an explanation (as well as the larger issue of how to present explanations at all). A sensible approach to choosing level of detail would be to let the individual using the system decide: initially a high-level or vague explanation could be presented, which could then be made more detailed when and where the individual requests.

4.4 Conclusion

The intent of plan explanation is to aid decision makers in understanding or comparing complex plans, and particularly in understanding the interdependence of different parts of the plan—on each other, or on assumptions or conditions. Plan explanation could be approached from an informal angle (by allowing or requiring decision makers to annotate parts of a plan with rationale explaining their choices), or formally (by using a formal action model). In either approach, one of the major technical issues is filtering the amount of information presented in an explanation to highlight the most significant interactions and downplay or ignore the vast amount of insignificant detail.

5. A Brief Survey of Explanation in Bayesian Networks

Since we are interested in describing plans by Action Networks, which are based on Bayesian networks, it is appropriate to begin our study of explanation with a survey of existing work into explanation in Bayesian networks.

5.1 Early History

Much of the early work in Bayesian networks, and almost all of the work in explanation in Bayesian networks, has been done in the domain of medical diagnosis and decision-making. The impetus for explanation in Bayesian networks more or less began with the publication of [Teach & Shortliffe, 1981], in which the authors found that medical professionals were reluctant to use medical expert systems, and that the primary reason for their reluctance was a lack of understanding of how expert systems arrived at their conclusions.

Bayesian networks themselves represented an improvement in presentation over raw statistical information. Early work, such as [Jimison, 1980], emphasized the use of the structure of the network itself as a way of explaining a domain. The intuitive nature of Bayesian networks (at least when they are constructed according to causal reasoning) remains a strong selling point for their use.

The graphical structure of Bayesian networks only tells part of the story, however—the other part concerns the strength of the relationships between variables, and the complexity of their interactions. Many researchers have ranked evidence variables by the strength of their influence on a target variable.[5] The GLADYS system described in [Spiegelhalter & Knill-Jones, 1984], for example, computes the "weight of evidence," (which is defined to be $\log P(E|H)/P(E|\neg H)$, where E is the evidence and H is the target variable) for each evidence variable. If the weight of evidence for a particular evidence variable is positive, then that evidence increased the probability of the hypothesis, while if it is negative, it decreased the probability. By listing the evidence variables together with their individual weights, the user (the medical clinician) could see which evidence contributed to and which conflicted with the conclusion, and by how much.

In another early research effort, [Jimison, 1980] displayed the not only the probability of the target variable, but the variance, allowing users of the system to see how the variance dropped as more evidence was entered. Her system also showed which nodes in a network were

[5] Early literature used the term "explanation" to refer to many different things, but by 1984, usage had largely stabilized to mean "how the evidence influences the diagnosis."

"sensitive," where a sensitive node is one which, if the variance of its probability were reduced, could affect the final diagnosis (that is, change which diagnosis had the highest probability).

5.2 Suermondt

The most important work in explanation in Bayesian networks is unquestionably the dissertation work of H.J. Suermondt [Suermondt 1991, Suermondt & Cooper 1992, Suermondt 1992]. Suermondt's INSITE system appears to have been the first to use the graphical structure of the network to describe the flow of information from evidence to the target variable, at least in multiply-connected networks.

Suermondt's approach is divided into two phases: (1) identification of the most important evidence variables (from amongst those given), and (2) determination of the most influential chains of inference from evidence nodes to the target variable (the hypothesis or diagnosis).

In the first phase, identification of important evidence variables, INSITE ranks the impact of variables on the target in a manner similar to GLADYS, but with a few differences. One minor difference is that where GLADYS uses weight of evidence, INSITE uses cross-entropy.[6]

Far more significant is the difference in independence assumptions. GLADYS makes the simplifying assumption that evidence variables are independent given the diagnosis (which implies that the weight of evidence of each variable can be computed independently of other evidence variables). INSITE, in contrast, acknowledges that evidence variables may be dependent, and therefore ranks not only individual evidence variables, but also *subsets* of evidence variables. If E represents the total set of evidence given, let F be a subset of E, then Suermondt defines the *cost of omission* of F to be the cross-entropy between the probability of the hypothesis D with the entire evidence set E and the probability of D without the set of evidence F: $H^{-}(F) = H(P(D|E);P(D|E\backslash F))$. By looking at both F and its complement E\F, Suermondt decides whether the subset of evidence F is *necessary* and/or *sufficient* to explain the change from P(D) to P(D|E). Of particular interest are subsets of evidence that are necessary, sufficient, and *minimal:* no smaller set is both necessary and sufficient.

One output of the INSITE system is a listing of minimal necessary and sufficient subsets of evidence, as well as an indication of conflicting evidence (which is determined as a byproduct of the system by identifying instances where a subset of the evidence variables creates stronger evidence for the hypothesis than does the full set of variables). Suermondt notes that alone may be sufficient explanation, at least for domain experts, but that in many instances, more information about how the evidence affects the hypothesis is desired. Thus the second phase of the system identifies those pathways in the network which are most important. In a multiply-connected network, this is difficult to define precisely. Quoting from [Suermondt, 1991]:

> "It is tempting to view chains between nodes as channels through which information flows. Most people familiar with belief networks can visualize the image of certain 'important' arcs, drawn as very thick arrows, which 'most of the information flows,' and others, drawn as thin arrows, that contribute only marginally to the transmission of evidence. Such an image, in which probabilistic updates are treated analogously to electrical currents, is overly simplistic and often invalid, especially when there are multiple direct chains from a finding E_i to the variable of interest D, which can occur when the network is multiply connected. For multiply connected networks, it is more difficult to determine how the evidence flows through the network. ... It is

[6] Cross-entropy is defined as $H(P(D); P'(D)) = \sum P(d) \log (P(d)/P'(d))$, where P and P' are two different distributions over a space D, and the summation is over the sample space of D. To measure the impact of an evidence variable E, the distributions are taken to be P(H) and P(H|E), for example. The two measures, weight of evidence and cross entropy, have subtly different properties, and which is the "right" one to use seems to be the source of some debate in the statistical community.

> difficult to predict definitively the combined effects of evidence transmission among multiple chains by analyzing the chains separately, since there are often poorly predictable interactions among chains."

The basic premise employed by Suermondt could be summarized by stating that he attempts to determine which chains (a chain is a complete path from an evidence node to the hypothesis node) are *not* transmitting evidence, and omits those chains from further consideration. He uses two techniques to do this: first, if there is some node along a chain whose probability does not change significantly when the evidence is added, then that chain cannot be significant. If a chain is judged significant, it can further be ranked by "cutting" it by removing one of the arcs in the chain (an arc that is not included in any other chain) and seeing how much the probability of the hypothesis is changed. Suermondt acknowledges that this approach is somewhat *ad hoc*, but the problem of understanding flow in multiply-connected networks is a difficult one.

5.3 Other Research

Citing evidence that humans are not particularly adept at probabilistic reasoning, [Henrion & Druzdzel, 1991], in work contemporary with Suermondt's work, seek to find forms of explanation that more closely resemble human reasoning. This leads them to using linguistic terms such as "highly probable" in place of numeric probabilities, and to emphasizing the necessity that models be structured to follow causality. They also suggest two techniques for generating explanations which are intended to follow human modes of reasoning.

The first is *qualitative belief propagation*, based on Wellman's qualitative reasoning model [Wellman, 1988]. (According to this model, the qualitative relationship between two variables is positive if, under all conditions, increasing the probability of one variable increases the probability of the other, and negative if, under all conditions, decreasing the probability of one increases the probability of the other; if neither condition holds, the relationship cannot be described qualitatively.) The basic idea is that if all the relationships in a network can be described qualitatively, then an explanation for the flow of evidence in a network can be given by tracing its qualitative path to the target variable. (The original paper described propagation in singly-connected networks; in [Druzdzel & Henrion, 1993] they extended the technique to multiply-connected networks). The restriction that all relationships be qualitative is a strong one, but many natural relationships do fall into this category (they particularly cite the NOISY-OR relationship, commonly used as a prototypical relationship for causal modeling).

The second explanation technique is based on *scenario generation.* The idea is to choose a few of the most likely scenarios (a scenario is an assignment to all relevant variables), and list them with their probabilities. Generally, there would be exponentially many possible scenarios, but Henrion and Druzdzel point out that often the few most probable scenarios contain most of the probability mass, giving a good overall picture of the possible cases. This technique can also display conflict, since some scenarios may disagree about the state of the target variable.

In more recent work, [Haddawy, et al.., 1997] describes a system called BANTER, which is intended for tutoring medical students. BANTER is largely based on Suermondt's INSITE system. BANTER adds verbal phrases to describe the relationships between variables; for example "X causes Y" or "X is detected by Y." These phrases are then used to construct natural language explanations which follow the chains of evidence flow (interestingly, unlike other researchers who have regarded the graphical structure of a Bayesian network as part of the explanatory power of their system, Haddawy, et al.., explicitly point out that the Bayesian network is hidden from the user—"In fact, nothing in the way the system interacts with the user would even indicate that the system is using a Bayesian network to perform its reasoning.") There are several other minor differences between INSITE and BANTER—for example, BANTER is concerned with determining which is the single next best test to perform, which leads them to revert to importance tests on single variables only (but conditioned on any evidence already present), and only for sufficiency, unlike INSITE's subset tests for both sufficiency and necessity. BANTER also handles non-binary variables by generalizing the measure of influence to be "positive", "negative" or "mixed", and can rank evidence by strength of influence in each category.

[Madigan, et al., 1997] presents another explanation system, but with the emphasis definitely on graphical visualization of information. As in previous research, Madigan, et al., are concerned with tracing the flow of influence from evidence to target variable through chains in the network. They display the strength of the relationship along arcs in the network by drawing the arc with two widths: the outer width indicating the maximum possible strength of information flow between variables (if one of the variables were instantiated directly, for example), and the inner width indicating the actual strength of the information actually flowing through the arc. They use color to indicate the direction of the relationship: blue for positive, red for negative. They solve the dependence problem by asserting an ordering to the evidence variables: the strength influence of evidence variables is not determined in isolation, but rather is conditioned on the evidence variables previously established (they provide a facility for reordering evidence variables so the user can explore their interdependencies).

The most significant technical aspect of Madigan, et al.'s approach is the way in which they define chains in multiply-connected networks. They prove that in a class of networks called Berge networks, which include some multiply-connected networks, a single chain is sufficient to completely explain the impact of evidence on the target node. When a network is not a Berge network, they allow the user to collapse variables until a Berge network has been created. In the collapsed network, single variables will be tuples of variables from the original network, which could make the explanation unwieldy, but it is better than collapsing the network until it is singly connected, and unlike previous work, the technique is sound.

5.4 Commentary

We are interested in explanation of plans, especially large-scale plans, rather than diagnostic networks. How would this affect the kinds of techniques discussed above?

The first apparent distinction is that diagnostic problems are represented by Bayesian networks, whereas decision problems are commonly represented by influence diagrams, which introduce choices and utilities into the model. However this difference is not really significant: [Shachter & Peot, 1992] shows a technique for a transforming an influence diagram into an equivalent Bayesian network. Moreover, if a particular plan or policy is given, then the algorithm for tracing the influence from decisions to utilities in that plan is identical to the algorithm for tracing the influence from evidence to hypotheses in a Bayesian network.

Explanation in diagnostic systems has generally been interpreted to mean explaining only one thing: the shift in probability of a target variable in response to the addition of evidence. In planning, we may be interested in shifts of probability, or we may be interested in shifts of utility (or more generally, in any measure of plan quality). And instead of adding evidence, we are more likely to be interested in comparing different plans, or hypothesizing different situations. This difference is also superficial, however: the techniques used in explanation of evidence in diagnosis really only depend on comparing two differing probability distributions; whether they differ because the evidence has changed, or because plans are different, is immaterial. (This does assume that measures of plan quality can be interpreted as utility models (that is, as preferences over outcomes); if this is not the case, then the situation may well be more difficult.)

More fundamental concerns are the adequacy of the modeling language presented by Bayesian networks, the appropriateness of the independence assumptions used in measuring strength of impact, and scalability.

Bayesian networks provide a “flat” description of a domain, where planning is almost always thought of as a hierarchical process, wherein certain decisions or activities are carried out subordinate to other higher-level decisions or activities. For an explanation system to be useful, it will be necessary to take hierarchy into account. There are currently several researchers investigating the incorporation of hierarchy into probabilistic models, but this is preliminary work, and at present we are unaware of anyone who has attempted bring explanation into the picture.

If used, the assumption that evidence variables (or more generally, any variables of interest) have independent effects on the target variable (or utility measure), will have to be justified in planning domains. A priori, it seems that there will be some situations in which it is justified (for example, the monetary cost of a plan is simply the sum of the independent cost of its parts) but many others in which it is not (for example, .scheduling decisions that are made both to accomplish a particular goal and to avoid conflict with other goals—the very foundation of AI planning has been that such interactions are unavoidable). If we cannot justify this assumption, then we will have to use something like Suermondt’s cost of omission measure over subsets of variables; the difficulty with this measure is that the number of subsets increases exponentially with the number of variables, which leads quickly to intractability. (In the next section, we describe a heuristic approach to this problem.)

Finally, the issue of scalability in general: the diagnostic models used for explanation in research in the literature have typically been small, on the order of tens of nodes, whereas the kinds of plans we anticipate seeing are much larger. Except for the techniques of Henrion and Druzdzel, every algorithm in this section is exponential (or doubly-exponential), and thus straight-forward application to larger systems will simply not be tractable. Moreover, even if the techniques were computationally adequate, the explanations generated would be in danger of becoming incomprehensible. In order to address this issue, new techniques or heuristic approximations to existing techniques will need to be found. In the long term, one of the most promising avenues of attack is to make use of the hierarchy in a hierarchical plan, using the higher levels of abstraction to “shrink” the size of the model whenever doing so does not lose too much information.

6. References

[Druzdzel & Henrion, 1993] Druzdzel, Marek J., and Henrion, Max, Efficient Reasoning in Qualitative Probability Networks, in *Proceedings of AAAI-93*, pp. 548-553.

[Haddawy, et al.., 1997] Haddawy, Peter, Jacobson, Joel, and Kahn, Charles E., Jr., BANTER: A Bayesian Network Tutoring Shell, to appear in *Artificial Intelligence in Medicine,* 1997.

[Henrion & Druzdzel, 1991] Henrion, Max, and Druzdzel, Marek J., Qualitative Propagation and Scenario-Based Schemes for Explaining Probabilistic Reasoning, in *Uncertainty in Artificial Intelligence 6,* P.P. Bonisonne, M. Henrion, L.N. Kanal and J.F. Lemmer, eds., Elsevier Science Publishers, 1991.

[Jimison, 1990] Jimison, Holly B., Generating Explanations of Decision Models Based on an Augmented Representation of Uncertainty, in *Uncertainty in Artificial Intelligence 4,* R.D. Shachter, T.S. Levitt, L.N. Kanal, and J.F. Lemmer, eds., Elsevier Science Publishers, 1990.

[Madigan, et al.., 1997] Madigan, David, Krzysztof, Mosurski, and Almond, Russel G., Graphical Explanation in Belief Networks, to appear in *Journal of Computational and Graphical Statistics*, 1997.

[Pearl, 1990] Pearl, Judea, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kauffman, 1990.

[Shachter & Peot, 1992] Shachter, Ross D., and Peot, Mark A., Decision Making Using Probabilistic Inference Methods, in *Proceedings of UAI-92*, pp. 276-283.

[Spiegelhalter & Knill-Jones, 1984] Spiegelhalter, David J., and Knill-Jones, Robin P., Statistical and Knowledge-based Approaches to Clinical Decision-support Systems, with an Application in Gastroenterology, in *Journal Royal Statistical Society A,* vol. 147, pp. 35-77, 1984.

[Suermondt, 1991], Suermondt, H.J., Explanation of Probabilistic Inference in Bayesian Belief Networks, Report KSL-91-39, Knowledge Systems Laboratory, Medical Computer Science, Stanford University, June 1991.

[Suermondt & Cooper, 1992] Suermondt, H.J., and Cooper, Gregory F., An Evaluation of Explanations of Probabilistic Inference, in *Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care,* pp. 579-585, November, 1992.

[Suermondt, 1992] Suermondt, H.J., *Explanation in Bayesian Networks,* Ph.D. Thesis, Stanford University, 1992.

[Wellman, 1988] Wellman, Michael P., *Formulation of Tradeoffs in Planning under Uncertainty,* Ph.D. Thesis, MIT Lab for Computer Science, 1988.